

Introdução à Modelagem Orientada a Objetos

Sistemas UML

Faculdade GranTietê

Prof. José Antonio Gallo Junior

Introdução à Modelagem de Software

Modelagem de software é o processo de criar representações abstratas da estrutura e comportamento de um sistema antes de sua implementação.

Por que modelar?

- Reduz erros na fase de desenvolvimento.
- Melhora a comunicação entre equipe técnica e stakeholders.
- Ajuda no entendimento e manutenção do sistema.

Analogias para facilitar a compreensão

- A modelagem de software é como **um projeto arquitetônico** antes da construção de um prédio.
- Imagine tentar construir um prédio **sem uma planta**.

Dessa forma a modelagem permite visualizar e estruturar o sistema antes do desenvolvimento.

Prática

“Se você fosse criar um **aplicativo de pedidos para um restaurante**, quais elementos (usuários, processos, dados) precisariam ser considerados?”

Fundamentos da OO

- **Objetos:** Entidades que possuem atributos e comportamentos.
- **Classes:** Modelos que definem os objetos.
- **Encapsulamento:** Esconder detalhes internos e expor apenas o necessário.
- **Herança:** Criar novas classes baseadas em outras
- **Polimorfismo:** Um mesmo método pode ter diferentes implementações dependendo da classe

Exemplo de Classe em Código (C#)

```
class Carro {  
    string cor;  
    string marca;  
    int velocidade;  
    void acelerar() { velocidade += 10; }  
    void frear() { velocidade -= 10; }  
}
```

Discussão

Como esses conceitos podem ser aplicados em um sistema real?

O que é UML?

UML (Unified Modeling Language) é uma linguagem gráfica para representar estruturas e comportamentos de um sistema de software.

- Padrão internacional para modelagem de software.
- Ajuda na documentação e comunicação entre equipes.
- Utilizada por desenvolvedores, analistas e arquitetos de software.

Principais Diagramas UML

Diagramas Estruturais (representam a organização do sistema):

- Diagrama de Classes,
- Diagrama de Objetos,
- Diagrama de Pacotes,
- Diagrama de Componentes.
- Diagrama de Implantação.
- Diagramas Comportamentais: Sequência, Atividades, Casos de Uso.

Principais Diagramas UML

Diagramas Comportamentais (representam interações e fluxos):

- Diagrama de Sequência.
- Diagrama de Casos de Uso.
- Diagrama de Atividades.
- Diagrama de Máquinas de Estado.

Atividade Prática

Identifique classes principais, atributos e métodos para criação de um sistema para gerenciamento de uma escola.

Conclusão da Aula

A modelagem ajuda no desenvolvimento estruturado de software.

A UML é uma ferramenta essencial para documentação e design.